REMARKS/ARGUMENTS

Claims 1, 3-10, 12-19, and 21-24 are pending in the present application. Claims 10, 12-19, and 21-24 were canceled; claim 1 was amended; and no claims were added. Reconsideration of the claims is respectfully requested.

Applicants wish to thank Examiner Nguyen for participating in a phone conference on July 23, 2008. Proposed amendments to claim 1 were discussed. Examiner Nguyen provided some guidance as to what may be needed to be added to the claims to achieve allowance. In accordance with Examiner Nugyen's guidance, Applicants have amended claim 1 in order to overcome the prior art.

Applicants have amended some claims and canceled others. Applicants do not concede that the subject matter encompassed by the earlier presented claims is not patentable over the art cited by the Examiner. Applicants canceled and amended claims in this response solely to facilitate expeditious prosecution of this application. Applicants traverse all rejections and respectfully reserve the right to pursue the earlier-presented claims, and additional claims, in one or more continuing applications.

I. 35 U.S.C. § 102, Anticipation

The Final Office Action has rejected claims 1, 3-10, 12-19, and 21-24 under 35 U.S.C. § 102(b) as being anticipated by *McMillan et al.*, U.S. Patent No. 6,118,448 (hereinafter "*McMillan*"). This rejection is respectfully traversed.

Regarding this rejection, the Final Office Action states:

As per claims 1, 10, and 19:
McMillan teaches:

obtaining the coverage data containing instruction access indicators associated with the code, wherein each instruction access indicator is associated with a *different portion* of *the code, and wherein each instruction access indicator is initialized* as *being unset prior to execution* of *its associated code portion* (see at least col. 4:33-40 "In the presently preferred embodiment the runtime engine 32 communicates with the integrated development environment through message passing. The message passing technique is used to communicate state information between the runtime engine and the integrated development environment. This state information is then used to impart different visual representations to the graphical objects displayed on the monitor work station 26"; see also col. 5:24-31 "the runtime engine 32 maintains a data store of state data 56. These state data record the runtime performance of the control program, by maintaining a record of how many times each program block was executed during a predetermined number of program cycles. if desired, state data may be recorded for each control block and each decision block that makes up the control program");

identifying instruction access indicators that have been set by a *processor in the data processing system in response to execution* of *the code by the processor to form set instruction access indicators, wherein each set instruction access indicator is associated with an executed portion* of *the code* (see at least FIG. 4; see also at least col. 5:46-50 "From the always executed state 62, a transition is made to the sometimes-executed state

64 (yellow) if the block was NOT executed on the last scan. Otherwise, the always executed state 62 remains unchanged"): and

generating a *presentation for the coverage data, wherein each set instruction access indicators is identified in the presentation* (see at least FIG. 4; see also at least col. 5:1-5 "...providing a graphical display that imparts different visual representations to the graphical object", to reflect runtime performance of the control program. In an exemplary embodiment the display will be different colors to show which branches of the control program (a) always execute, (b) sometimes execute and (c) never execute"); and

identifying unset instruction access indicators that have remained unset during the execution of *the code by the processor, wherein each unset instruction access indicator is associated with an unexecuted portion* of *the code, and wherein each unset instruction access indicator is identified in the presentation* (see at least FIG. 4).

Final Office Action dated April 29, 2008 pp. 3-4 (emphasis in original).

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case, each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

Amended independent claim 1 recites:

1. (Currently Amended) A method in a data processing system of presenting coverage data for code, the method comprising:

receiving, by an instruction cache, code for execution;

responsive to the code being executed by an execution unit, sending a signal, by a completion buffer, to the instruction cache that the code has executed;

responsive to receiving the signal from the completion buffer, setting, by the instruction cache, in metadata, an instruction access indicator, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion;

obtaining the coverage data containing instruction access indicators associated with the code;

generating a presentation for the coverage data, wherein each set instruction access indicator is identified in the presentation; and

identifying unset instruction access indicators that have remained unset during the execution of the code by the processor, wherein each unset instruction access indicator is associated with an unexecuted portion of the code, and wherein each unset instruction access indicator is identified in the presentation.

McMillan teaches the use of software to track execution of portions of a program. As shown in Figure 3 and explained in column 4, line 23- column 5, line 33, program trackers track the execution of portions of executable code. Control programs are developed and compiled into executable code that is sent to a runtime engine. The runtime engine stores state information, which records the runtime performance of the executable code by recording how many times each program block was executed during a predetermined number of program cycles. The state information is communicated to the program trackers through a communication mechanism, such as an OLE mechanism. This information is then presented to a flowchart interface, which causes the information to be displayed on a display. As known to one of ordinary skill in the art, a runtime engine refers to a virtual machine that manages a program. Thus, *McMillan* teaches a software implemented method of tracking execution of portions of compiled code.

In contradistinction, claim 1 recites hardware implemented steps for generating code coverage data including "receiving, by an instruction cache, code for execution," "responsive to the code being executed by an execution unit, sending a signal, by a completion buffer, to the instruction cache that the code has executed," and "responsive to receiving the signal from the completion buffer, sending, by the instruction cache, in metadata, an instruction access indicator, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion." *McMillan* does not teach or even mention hardware components such as an instruction cache or a completion buffer or the features of "receiving, by an instruction cache, code for execution," "responsive to the code being executed by an execution unit, sending a signal, by a completion buffer, to the instruction cache that the code has executed," and "responsive to receiving the signal from the completion buffer, sending, by the instruction cache, in metadata, an instruction access indicator, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion."

Further, as stated by Examiner Nguyen in the conference call on July 23, 2008, *McMillan* does not teach that the instruction access indicators are in metadata associated with the executed code portion. Thus, *McMillan* does not teach "responsive to receiving the signal from the completion buffer, setting, by the instruction cache, in metadata, an instruction access indicator, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion."

Therefore, for at least the reasons set forth above, Applicants submit that the *McMillan* reference fails to anticipate claim 1, as the *McMillan* reference fails to teach all the features of claim 1. Since claims 3-9 depend from claim 1, the same distinctions between *McMillan* and the claimed invention in claim 1 applies for these claims as well. Claims 10, 12-19, and 21-24 have been canceled.

Therefore, the rejection of claims1, 3-10, 12-19, and 21-24 under 35 U.S.C. § 102(b) has been overcome.

II. **Conclusion**

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: July 29, 2008

Respectfully submitted,

/Gerald H. Glanzman/

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GG/blj